ISO 10303 STEP AP 233- Systems Engineering Response to San Francisco 1/99 Meeting Action Item:
"Detail further the USS"

Uniform System Semantics (USS) & Linguistic Variables

By Harold P. Frisch
NASA/Goddard Space Flight Center
PDES Inc.
March 2, 1999

Background

During the San Francisco STEP meeting of the AP 233 team it was proposed that there was a need to introduce into the current set of Units of Functionality (UoF) a capability for organizations to create a product-line specific "Uniform System Semantics" set. It was argued that:

- if AP 233's current set of UoFs enable the *unambiguous* representation of requirements, function and behavior,
- and if the semantics used for defining product measures is *ambiguous*,
- then the product's systems engineering data model will also be *ambiguous*.

It was also argued that in order to enable the creation of intelligent support aids product characterization variables must be described both quantitatively by numeric measures and qualitatively by linguistic measures. There was also a need to provide for the representation of the linguistic relations used to define rules, constraints and measures that cannot be represented by crisp deterministic mathematical expressions.

All team members in attendance felt that there was a need for the discussion to be expanded and documented in depth. This note satisfies to the associated action item.

Microscopic vs. macroscopic product views

The STEP effort seeks to develop standards for the representation and exchange of product data. The associated STEP Application Protocols (AP's) are accomplishing this in a variety of domains of interest. AP 203 and AP 214 are enabling communication of mechanical CAD data, AP 210 will do the same for electrical CAD data and AP 209 will do the same for engineering analysis of metallic and composite structures and other engineering analysis areas that need to interface with a structural model; e.g., computational fluid dynamics. To accomplish this a uniform semantics set is effectively established at the microscopic data level. AP 233 seeks to support systems engineering. This domain of interest needs to view products at a more macroscopic level.

When organizations have a project centric organizational structure communication across projects within a product line is frequently minimal and often very weak. Couple this with the need for geographically distributed collaboration within and across corporate and national boundaries and the need to utilize an extensive array of subcontractors, the unambiguous representation and exchange of product data becomes very difficult.

Current AP 233 Units of Functionality (UoFs)

AP 233 has currently developed UoFs to describe requirements, functional architecture, behavior, configuration management and graphics. Physical architecture is planned. These are generic and independent of product line. It is up to the AP 233 user's organization to harmonize the product characterization terminology.

Product Measures – Another virtual product view

The Uniform System Semantics set seeks to limit its focus to those terms used to describe product characterization and maturity measures. The union of these measures provides the systems engineer with a virtual view of the product and its development process. This view is sufficient to define the product's physical and functional architecture along with its cost, maturity, performance, operations and all other life cycle factors considered relevant to the systems engineer.

It is recognized a-priori that product measures change in value, relative importance and susceptibility to change through the life cycle stages of: propose, design, build, maintain and decommission.

This information is important to record and maintain. Quality improvement teams will need the ability to trace product measure values and their associated change rational back through the systems engineering process to understand performance, operations, maintenance and decommissioning problems relative to early design decisions.

Product Measures – Across a Product Line

Normally new products within a product line are formally proposed. These proposals contain the set of all measures need to unambiguously and completely describe the new product to the review team. This is an excellent place to start developing an USS set.

Proposals are normally written at a high level using semantics set that the proposing group expects to be readily understood to the review team. The proposal addresses requirements and allocates these to the physical and functional architecture of the product. It also considers all factors necessary to characterize product life cycle operation and costs. The union of all product characterization and maturity measures used within a sampling of product-line proposals provides an excellent beginning.

Uniform System Semantics (USS) - Harmonized Product Measures

To develop the USS set it is necessary to first assemble a set of product characterization and maturity measures. This list frequently exists as an informal spreadsheet of parameters that proposal development teams within a group use to guide their work. If the USS set is to support collaboration semantics harmonization between groups is essential. It is common for collaborating groups to define identical measures differently, combine them differently and present them in a manner that prevents transformation back to the original set of base measures.

The Concept of Measure

AP 233 seeks to capture more than just the ability to attach physical units and numeric values to the product measures that quantify product requirements. Knowledge representation, its utilization, rules, constraints, degree of interest, degree of relationship, measures of success, maturity, etc. all benefit from the ability to create executable expressions that use fuzzy linguistic measures within fuzzy linguistic relations.

This note proposes that AP 233 include an ability to assign both a numeric and a linguistic measure, along with their associated units of measure, to all product measures. Numeric measures will support business as usual. Linguistic measures and the linguistic relations that they can be used to form will enable an organization to define and link product-line knowledge to the design databases of its new products.

For example: "Change notification" is a most basic systems engineering need. A change that may be insignificant for one design group may be of critical importance to another. The ability for a design group to linguistically define a degree of interest measure (weak, strong, very-very strong) to a specific subset of product measures would enable change notices to be delivered where needed along with a linguistic estimate of importance and associated reasoning. In like manner cross-disciplinary design checks, rules-of-thumb, corporate product knowledge, etc. can be encapsulated via linguistic relations and linked to the product design databases. As new data enters the database and product measures change alert reports can be automatically generated and distributed where needed.

Product Requirements and Product Maturity

Product needs are cast within the framework of product requirements, which are in turn expressed in terms of desired value ranges for product characterization measures. These requirements are not strictly associated with just the engineering domain. They include cost, time, performance, operations and a variety of other measures that support the _ilities (testability, reliability, maintainability, manufacturability, etc.). Product maturity checks and measures are also provided to aid in the insurance that a quality product is being produced. These are extensively detailed within SE standard EIA 731 "Systems Engineering Capability Model (SECM)". It can be obtained at
http://www.geia.org/eoc/G47/g47.htm

The forward of EIA 731 relates its work to the standards EIA 632 "Processes for Engineering a System" and IEEE 1220 "Application and Management of the Systems Engineering Process" as follows:

> "*(These)* standards (*EIA 632 & IEEE 1220*) define the "what-to-do's" of the processes for engineering systems and this model *(EIA 731)* provides a basis of determining "how well" those processes are defined and implemented".

The "how well's" are capsulated by maturity measures. The "what-to-do's" define the processes used to insure that the product produced satisfies its requirements as quantified by the product characterization measures.

For additional information on the quagmire of systems engineering standards see *Systems Engineering Standards and Models Compared*, by the "Software Productivity Consortium", at http://www.software.org/pub/papers/9804-2.html

Another summary of the quagmire of systems engineering standards can be found at http://www.vtcorp.com/WMA-INCOSE/docs/1997andBefore/lakestds.pdf. This is a slide presentation entitled **Standards That Can Help the Engineering of Better Systems by** Dr. Jerry Lake, Systems Management international (SMi).

Derived Requirements and Associate Product Measures

At the product proposal stage product requirements are allocated to either the physical or the functional architecture as bounds on "high level" product characterization measures. Once work begins, a myriad of derived requirements evolve. These are also quantified in terms of more product characterization measures. At the derived requirements level, associated product characterization measures become even more ambiguous. It is very common for different groups within the same large organization to define, combine and use seemingly identical measures in very different ways.

After the successful harmonization of product measures, the data model can link them to the requirements set that they are used to quantify. Once the link to the requirements set is established, links to the functional and physical architecture can be made. Maturity measures are linked to the systems engineering processes and then to the product's functional and physical architecture under development.

Recognition of the obvious fact that derived requirements are derived, implies that relations and dependencies exist between associated product characterization measures. These relations and dependencies are rarely captured in any formal sense. This note proposes that AP 233 enable the capture or both mathematically crisp and linguistic fuzzy relations and dependencies.

AP 203 Parts vs. AP 233 Product Measures

The union of all product characterization and maturity measures provides the systems engineer with a virtual view of the product. In a sense, the systems engineer views each product measure as a part of the product.

"Parts" provide the design engineering team with a virtual view of the systems physical and functional architecture. In an analogous manner "product measures" provide the systems engineering team with a view of both architecture and process.

This perspective allows the "parts" information model created for AP 203 (and probably AP 214) to be become an information model template for AP 233 "product measure".

Direct plagiarism from Section 2.8 of the PDES Inc. "Recommended Practices for AP 203" document leads to the following "product measure" informaton elements:
(download link at NASA STEP Central - website http://misspiggy.gsfc.nasa.gov/step/)

- Both parts and product measures have associated **description**s.
  - Parts have **description,** entered as a text string
    - Product measures also need a text string **description.** They also need:
      - Greco-Roman mathematical symbols used both stand-alone and within

    mathematical equations. Frequently the only way to remove the possibility of ambiguity is to include the defining mathematical expression within the **description**.
    - Graphs and block diagrams also need to be used at times for the removal of ambiguity in the **description**.

- Parts have a **name** and a unique **id**.
  - Product measures have unique **name**s
  - Product measure **id**'s can be non-unique. They are normally expressed as:
    - Acronyms derived from their name
    - Alphanumeric strings associated with output from a computer program,
    - Greco-Roman mathematical symbols
  - **id** uniqueness for product measures is NOT global. Different engineering domains often use the same symbol for very different meaning. It is hard to find an engineering domain that does not use (i, j e, k, λ, σ, ω).

- Parts have **version**s
  - Product measures also have **version**s.
    - Versioning enables life cycle traceability of product measure change.
    - Product measures can also be linked to life cycle stage:
      - Analysis
      - Design
      - Digital pre-assembly
      - As-Built
      - As-Maintained

- Parts are associated with a **person_and_organization** in the role of creator and controller of "change" and **approval** authority.
  - Product measures should also be associated with a **person_and_organization** having the role of creator, controller of "change" and **approval** authority.

- The **product_definition** entity establishes specific life cycle stage view of the product data: Analysis, Design, Digital pre-assembly, As-Built, As-Maintained. It also establishes relations such as: part to part and part to shape.
  - Product measures need the same.
    - Product measures normally cannot be viewed as independent variables. There are obvious and subtle dependencies. When dependency relations are non-analytic, linguistic measures for estimating **degree_of_relation** can become useful for intelligent support aids.
    - Product measures also need to be related to both **numeric_value** and **linguistic_value** along with their associated **units_of_measure**; just as parts are related to shape in AP 203.

- AP 203 provides for a **product_category** that can be used to group parts into general categories and for creating hierarchical networks of categories. Categories can be extremely useful in adding intelligence to the data.
  - Product measures need the same. Suggested categories are:

    - Engineering
    - Cost
    - Maturity
    - _ility
    - Performance
    - Operations
    - Maintenance
  - Each of these can be further broken down as needed.

- AP 203 provides for relating parts to specifications, contracts, supplier
  - Product measures need to be related to requirements

The above list of part vs. product measure analogies provides and adequate starting point for using AP 203 to support a product measure information model capability in AP 233.

Linguistic Variables, Measures & Relations

The book "Fuzzy and Neural Approaches in Engineering" by L.H. Tsoukalas & R.E. Uhrig, Wiley Interscience, 1997, ISBN 0-471-16003-2 contains, in my opinion, a most readable presentation of the titles subject matter. Many of the following definitions were taken directly from this reference (source page noted).

To motivate the development of an information model for linguistic variables, measures and relations the following definitions are presented:

- The theory of possibility is analogous and yet conceptually different from the theory of probability [pp. 15 & 38].
  - Possibility theory attempts to quantify how accurately a sample resembles an ideal element of the population.
  - Possibility theory focuses more on the imprecision intrinsic in language, whereas probability theory focuses more on events that are uncertain.
  - Other types of fuzzy measures are:
    - Belief measures
    - Plausibility measures
    - Necessity measures
    - Probability measures
  - *Fuzzy set* theory uses the concept of a *membership function*. The degree of membership within a set is indicated by a number in the interval [0,1]. [pp 15]
    - A membership function maps every element of the universe of discourse *X* to the interval [0,1].
    - Membershihp functions are a simple yet versatile mathematical tool for indicating flexible membership to a set and for modeling and quantifying the meaning of symbols.
    - All *fuzzy value* symbols (hot, long, heavy, …) will have an associated membership function.
      - Membership functions are primarily subjective in nature; this does not

mean that they are assigned arbitrarily, but rather on the basis of application-specific critera.

- Possibility is a *fuzzy measure*, which means that possibility is a function with a value between 0 and 1, indicating the degree of evidence or belief that a certain element *x* belongs to a set. A possibility of 0.3 for element *x*, may indicate a 0.3 degree of evidence or belief that *x* belongs to a certain fuzzy set.
  - If the application domain is human adult height and the fuzzy set is “tall” the element *x* might have been 5.5 feet. If the population group is restricted to just Japanese female adults the membershhip function would have to be changed.

- Fuzzy linguistic descriptions (often called *fuzzy systems* or simply *linguistic descriptions*) are formal representations of systems made through fuzzy *if/then* rules [pp. 105].
  - Fuzzy linguistic descriptions have rigorous mathematical foundations involving fuzzy sets and relations. They encode knowledge about a system in statements of the form

    if (a set of conditions are satisfied)
    then (a set of consequences can be inferred)
- A *linguistic variable* is a variable whose arguments are fuzzy numbers (and more generally words modeled as fuzzy sets), which we refer to as *fuzzy values* [pp. 106].
- Individual *if/then* rules are connected with the connective *ELSE* to form a *fuzzy algorithm.* Propositions and *if/then* rules in classical logic are supposed to be true or false. In fuzzy logic they can be true or false to a degree [pp. 106].
- Despite the difference in appearance, linguistic and conventional (analytical) descriptions are in fact equivalent to each other. Both can be used to describe the same system [pp. 107].
- The process of evaluating a fuzzy linguistic description is called *fuzzy inference*.

In summary:

> Describing a system through a linguistic description, no matter for what purpose, involves specifying in some way *linguistic variables*, *if/then* rules, and evaluation procedures known as *fuzzy inference* [pp. 113].

This statement defines the essential elements for a fuzzy information model The model must contain all information necessary to enable the operations of *fuzzy inference.* The evaluation procedures of *fuzzy inference* are out of scope for AP 233.

Fuzzy Information Model

The following information chain must be captured:

- Each "product measure" can be identified as a linguistic (fuzzy) variable.
- Sets of *fuzzy values* need to be defined that will cover the full range of product measure categories and subcatagories
  - Primary values may be non-dimensional:
    - Low, medium, high
    - Weak, medium, strong
    - Possibly true, neutral, possibly false

- Or they may be dimensional
  - Cold, medium, hot
  - Short, medium, long
  - Light, medium, heavy
- To increase granularity linguistic modifiers can be used, such as [pp. 117]:
  - Very
  - Very-very
- Many very different product measures may use the same fuzzy values.
  - The associated membership functions will be product measure unique and provision must be made to allow for the membership function to change through the product's life cycle.

- Knowledge is encoded in fuzzy linguistic descriptions of the form [pp. 105]

  if (a set of conditions are satisfied)
  then (a set of consequences can be inferred)
  - The information model must provide the capability to capture the:
    - *if_set* - if this set of conditions is satisfied
    - *then_set* - then this set of consequences that can be inferred
  - The process of evaluating a fuzzy linguistic description is call *fuzzy inference*.
  - *Fuzzy inference* is out-of-scope for AP 233.

- *Fuzzification* is used to transform a crisp set into a fuzzy set [pp. 25]. This process may be automated via a fuzzifier function. The fuzzification process is out-of-scope for AP 233.
- *Defuzzyfication* is used to transform the fuzzy output of a fuzzy inference process to a crisp number within the low and high bounds of the domain of discourse [pp. 163]. The defuzzyfication process is out-of-scope for AP 233.

- Capturing low and high range numeric limits for the universe of discourse an associated granularity measure would be in scope for AP 233. Defining and applying a fuzzifier function or defuzzyfication process is out of scope.

A Cerebromorphic Aided Systems Engineering Process

A Cerebromorphic aided systems engineering process supports, enhances and emulates the cerebral process of systems design and system design optimization. The proposed additions to AP 233 provide the information modelling infrastructure need to enable the creation of such a capability.

To form a judgement, measures are created in one's cerebral processor for the existing problem. These are then compared with associated measures of recored from past experience. Ambiguity can exist, and it is manifested by a state of cerebral confusion. This occurs when the measures of past experience do not obviously relate to the measures being associated with the current problem. Past experience is normally cerebrally recorded as qualitative measures associated with some aggregation of experienced-based rules, relations and constraints. Judgements are formed by first eliminating confusion. Asking the questions that will remove ambiguity does this. With ambiguioty removed, the new set of unambiguous measures are compared to the set of past experience measures. Somehow the mind aggregates everything together to produce a qualitative judgement that is presented as a linguistic measure.

This proposal has sought to provide the enabling infrastructure to support, enhance and emulate this generic process.

- The Unified System Semantics (USS) set provides unambiguous definitions of all product characterization measures.
- Linguistic (fuzzy) measures provide the ability of attach qualitative measures to the elements of the USS set.
- Membership functions provide a mathematical link between fuzzy measures and relevant numeric ranges for associated product characterization measure
- Linguistic *if/then* relations provide the ability to capture rules, relations and constraints in a cerebrally natural manner. The totality of these forms a product line specific knowledge base.
- Fuzzy inference (out of scope for AP 233) methods allow the linguistic relations to be aggregated to yield a linguistic result.

The envisioned Cerebromorphic aided systems engineering process continuously evaluates changes in the system characterization parameters relative to the knowledge base. When the fuzzy inference capability obtains a result that indicates a potential problem exists an alert notice is automatically sent to the owners of the associated measures along with a degree of importance estimate and associated rational attached.